Web Applications for Modern Control Systems

Beau Harrison

Controls Modernization

10th April 2019

Benefits of Browser-Based Applications

- Familiarity
  - Tabbed browsing
  - Scroll and zoom
  - Resizing
  - Navigation between pages
- Accessibility
  - Zoom
  - Specialized keyboards
  - Screen readers
  - High-contrast
- Standards
  - HTML (1993)
  - CSS (1996)
  - JavaScript (1995)
- Support
  - Google (Chrome)
  - Mozilla (Firefox)
  - Apple (Safari)
  - Microsoft (Edge)
- Rapid Development

Potential Downsides

- Volatility of the platform
  - JavaScript has never been volatile, ECMAScript standard
  - New web app development tools are created everyday
    - We shouldn't adopt the newest thing
    - Stay close to the ECMAScript standard
- Performance
  - Strategies for offloading large workloads
    - Server-side service
    - Browser services workers, separate thread e.g. squoosh
    - Web Assembly, systems code run in the browser e.g. WebP
  - Graphics
    - Canvas API for 2D
    - WebGL API for 2D and 3D uses graphics hardware

Known Downsides

- Design flexibility
  - Application design is no longer constrained
  - We must instate standards for web applications
  - Potential for non-conformant applications
- New
  - Build and deployment tools
  - Development practices and tools

Benefits of Choice

- Component-based design
  - Modularity
  - Reusability
  - Small code
  - Ease of use
  - Visual and functional consistency
- React
  - Fast visual updates
  - Rich customizable tools
  - Active community

Visions for the Future

- Central Component Repo
  - In the works with multiple modules already deployed
  - Standard set of tools to encourage application uniformity
- Central Build Environment
  - Adopt modern continuous integration/deployment standards
- Drag and Drop App Builder
  - Investigating GrapesJS
- Adoption of Web Assembly for reuse of C code
  - Tech is still in infancy
  - Autodesk is using Web Assembly for web viewer

Demos